Movie Recommendation Engine

Recommendation engines are an automated form of a “shop counter guy”. You ask him for the product. Not only he shows that product, but also the related ones which you could buy. The ability of these engines to recommend personalized content, based on past behavior is incredible. It brings customer delight and gives them a reason to keep returning to the website.

Client

Many online businesses rely on customer reviews and ratings. Explicit feedback is especially important in the entertainment and ecommerce industry where all customer engagements are impacted by these ratings. Netflix relies on such rating data to power its recommendation engine to provide the best movie and TV series recommendations that are personalized and most relevant to the user.

Data Set: The MovieLens DataSet

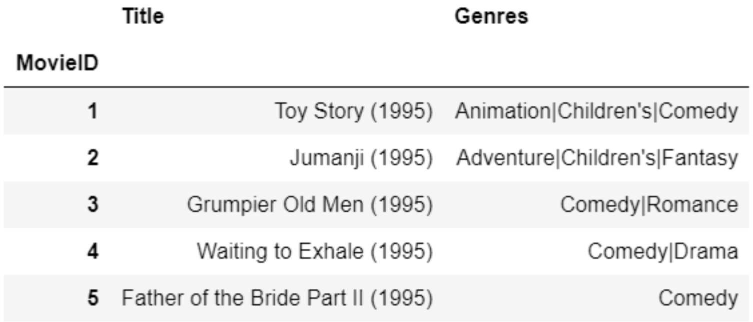
We will be using the MovieLens dataset for this purpose. It has been collected by the GroupLens Research Project at the University of Minnesota. MovieLens 100K dataset consists of:

- **100,000 ratings** (1-5) from 943 users on 1682 movies.
- Each user has rated **at least 20 movies.**
- Simple demographic info for the users (age, gender, occupation, zip)
- Genre information of movies

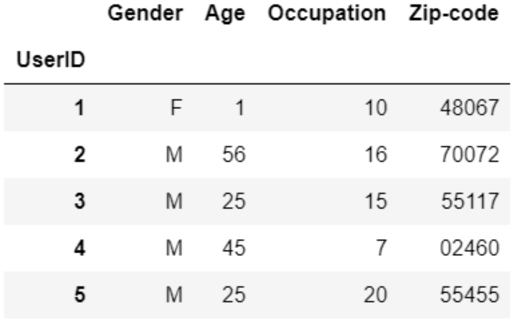
Data Wrangling

First, we input data & do data wrangling. The primary data source is MovieLens dataset which includes 3 .dat file for movies, users and ratings. We loaded each dataset to different dataframe.

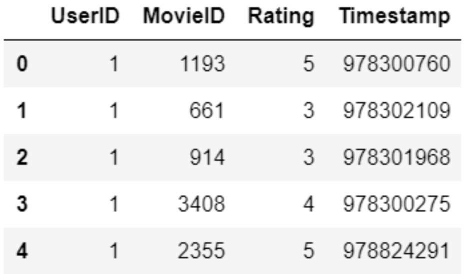
We see first five rows of each data frame:



	Title	Genres
MovieID		
1	Toy Story (1995)	Animation|Children's|Comedy
2	Jumanji (1995)	Adventure|Children's|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	Comedy|Drama
5	Father of the Bride Part II (1995)	Comedy



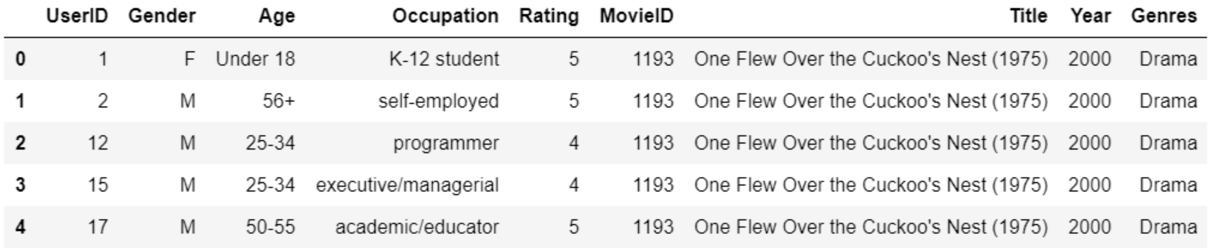
	Gender	Age	Occupation	Zip-code
UserID				
1	F	1	10	48067
2	M	56	16	70072
3	M	25	15	55117
4	M	45	7	02460
5	M	25	20	55455



	UserID	MovieID	Rating	Timestamp
0	1	1193	5	978300760
1	1	661	3	978302109
2	1	914	3	978301968
3	1	3408	4	978300275
4	1	2355	5	978824291

We conduct the following steps to clean up data and pick useful features.

1. Merge all 3 tables to one table and mapped each column which has code to real data and picked useful columns for further analysis



	UserID	Gender	Age	Occupation	Rating	MovieID	Title	Year	Genres
0	1	F	Under 18	K-12 student	5	1193	One Flew Over the Cuckoo's Nest (1975)	2000	Drama
1	2	M	56+	self-employed	5	1193	One Flew Over the Cuckoo's Nest (1975)	2000	Drama
2	12	M	25-34	programmer	4	1193	One Flew Over the Cuckoo's Nest (1975)	2000	Drama
3	15	M	25-34	executive/managerial	4	1193	One Flew Over the Cuckoo's Nest (1975)	2000	Drama
4	17	M	50-55	academic/educator	5	1193	One Flew Over the Cuckoo's Nest (1975)	2000	Drama

2. Search for missing values and filling them with appropriate value like mean
   - Didn’t find any missing value
3. Finding outliers & Fix then
   - Trying to find outliers and check the values if they are in the scope or not? For example, “Ratings” data frame, has one column ‘Rating’ which must have a value between 1 to 5, and apply command “.groupby(‘Rating’)” to data frame and check this column has any other value other than 1 to 5 or not? If yes, replace with appropriate value.
4. Drop duplicates
   - Searching for duplicate rows in data frames and if find any, investigate more to figure out what strategy must take (remove or keep). In the “User” data frame, I found duplicated rows which has different “UserId” but the other columns(‘Gender’ , ’Occupation’ , ’Zip Code’) are the same and it sounds to me, they are the same users, so for more investigation I merged “Users” data

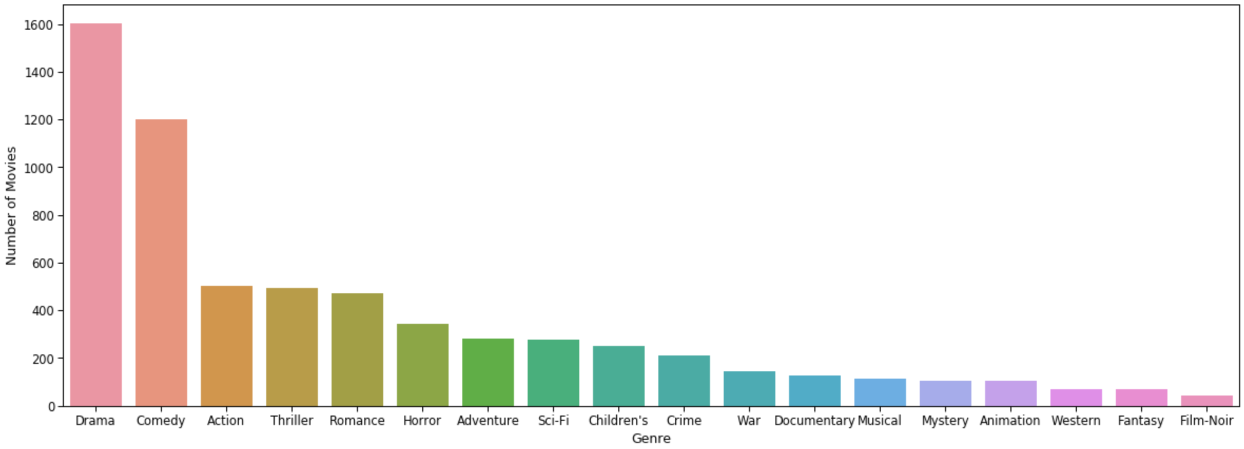
frame with “Ratings” data frame to correlate the users with their ratings and then perform grouping to see how many rows exist for each of these users (which they seem the same users) and figured out each of these user has a lot of ratings, so I had to decide to keep these rows as they are or delete duplicated rows from users and just keep one of them and merged all related rows in Ratings data frame and assigned them to that one existent user. After thinking about the final result of this project, I came up with the solution to keep them as they are like different users, because the existence of these duplicated users doesn’t have any side effects on any final reports and analysis

Data Exploration

we ask the following questions:

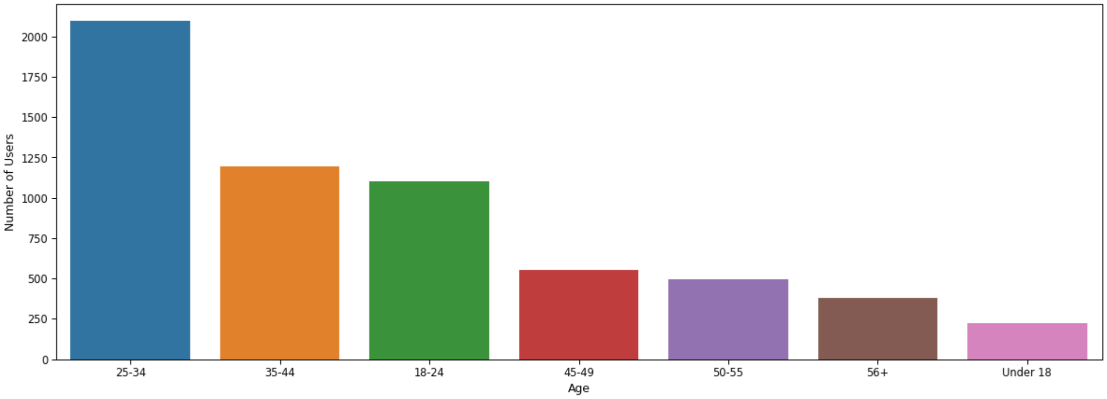
1. What might be the most important features that affect number of ratings and average of ratings?
2. Which age group has more affect on ratings?
3. Distribution of ratings on each factor?

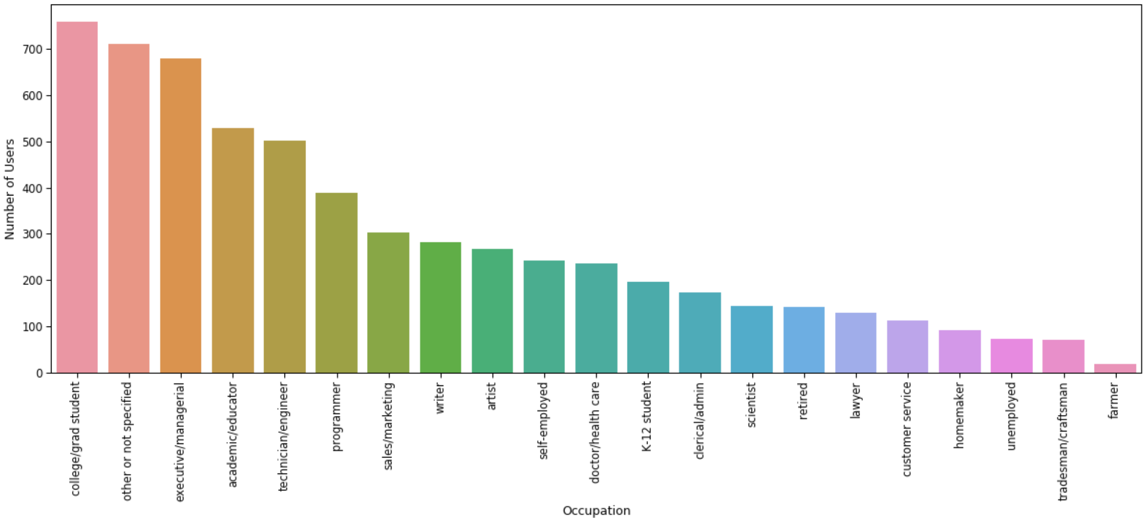
- Distribution of movies in each genre



As we see, 50% of ratings are belongs to 2 genres ‘Drama’ and ‘Comedy’.

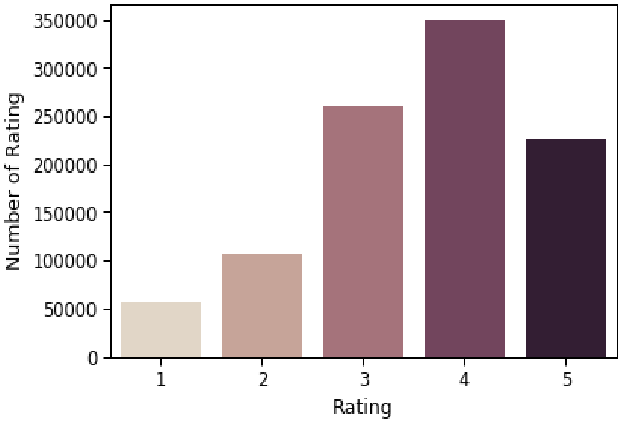
- Distribution of users in each age group and in each occupation

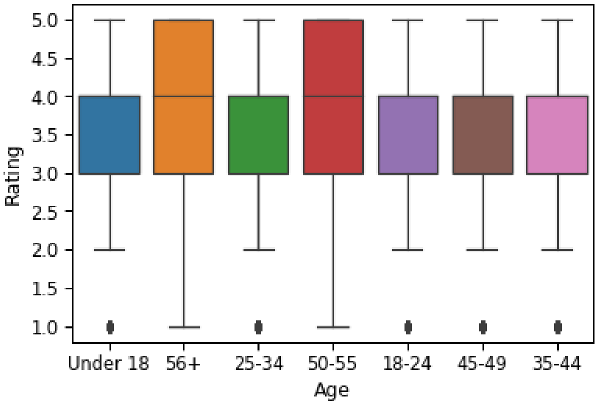




Most of users are between 18 to 34 years old and their occupation are one of these categories: Student, Executive, engineer or programmer

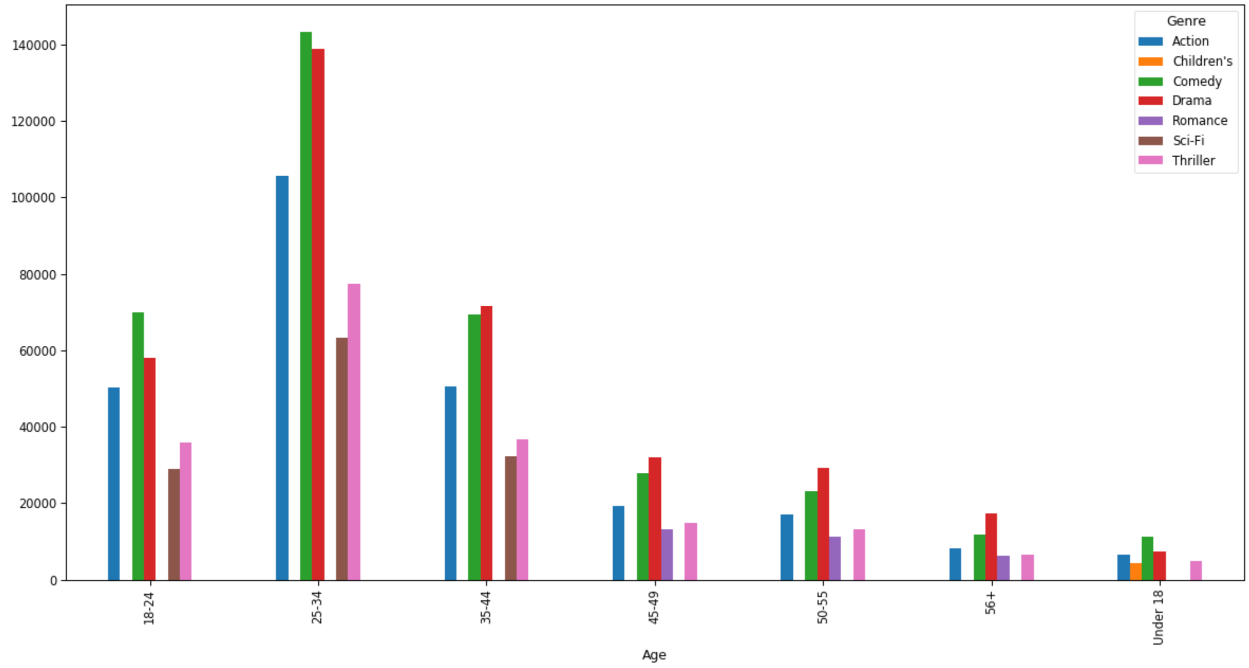
- Distribution of ratings





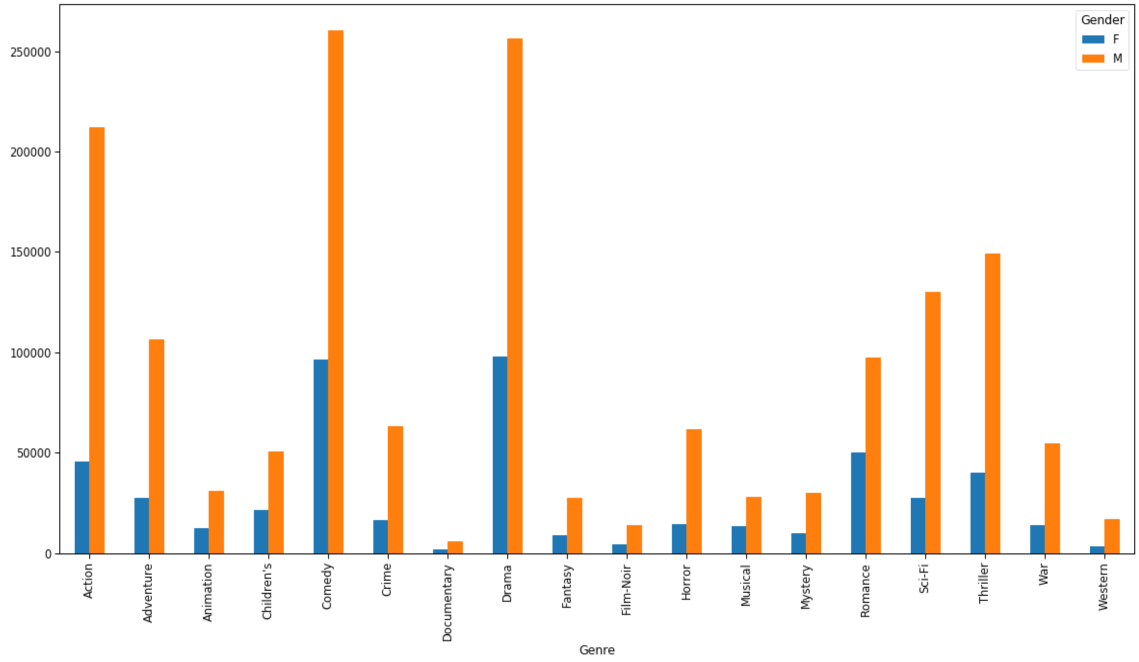
Portion of rating '4' is more than the other ratings and second box plot shows most of age groups has rating between '3' and '4'.

- Distribution of rating of most 5 popular genres in each age group

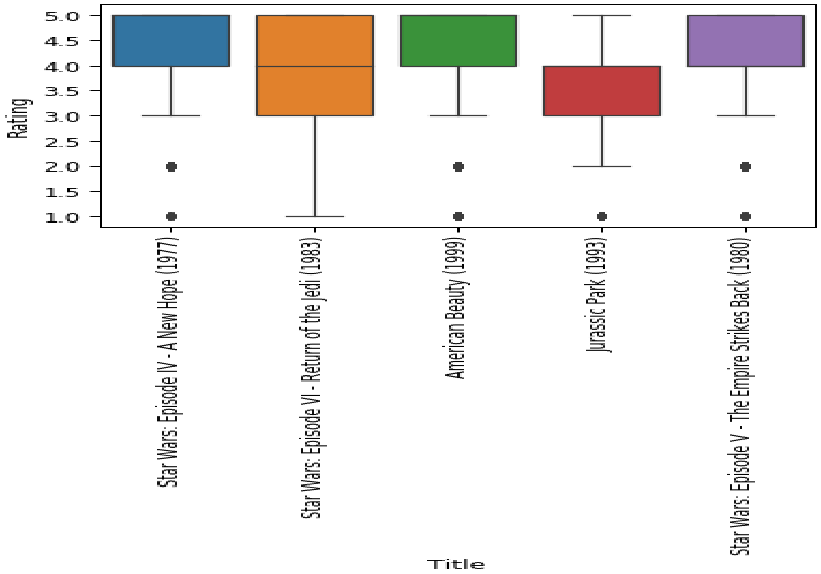


As we see, Comedy and Drama are most popular genre in each age group.

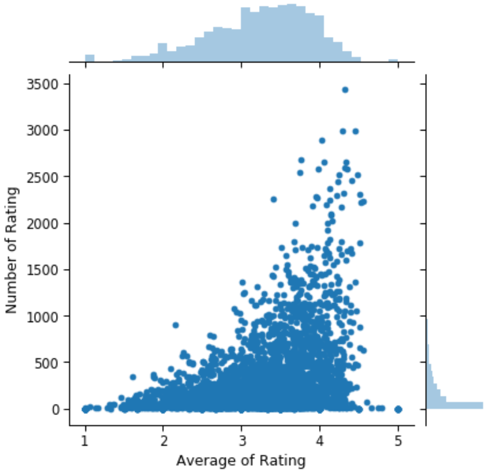
- Comparison of rating of each genre in each gender



- Distributation of rating for top five movies in dataset



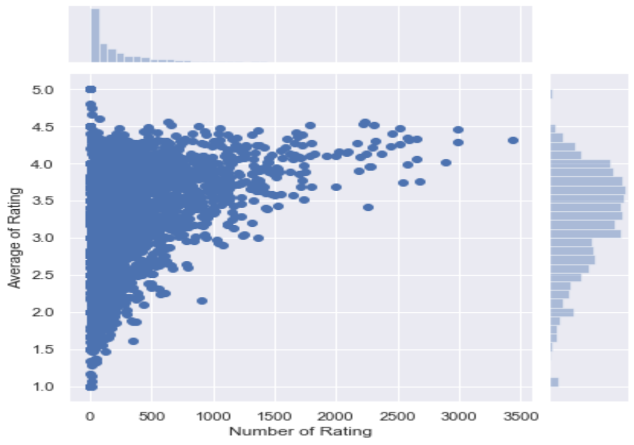
- Correlation between number of ratings and average of ratings

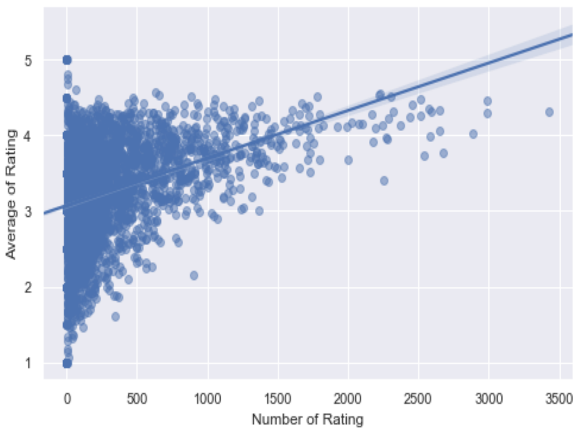


EDA – Inferential Statistics

In the following EDA, we try to perform analysis to find correlation between Number of Ratings and Average of Ratings for each movie.

Initial observations for Correlation between Number of Ratings and Average of Ratings





Pearson correlation coefficient is 0.35, not equal zero and positive, which means when number of ratings increases, Average of Ratings increase, but the correlation is pretty weak.

Setup hypothesis test

Ho : The Average of Ratings and Number of Ratings are independent

Ha : The Average of Ratings and Number of Ratings are correlated

Statistical significance for α = 0.01

Test Statistic: Pearson correlation coefficient

To do so, permute the Number of Ratings but leave the Average of Ratings values fixed. This simulates the hypothesis that they are totally independent of each other. For each permutation, compute the Pearson correlation coefficient and assess how many of our permutation replicates have a Pearson correlation coefficient greater than the observed one.
As p-value less than α, we can reject the null hypothesis and accept alternative hypothesis which means that there is a correlation between number of ratings and average of ratings.

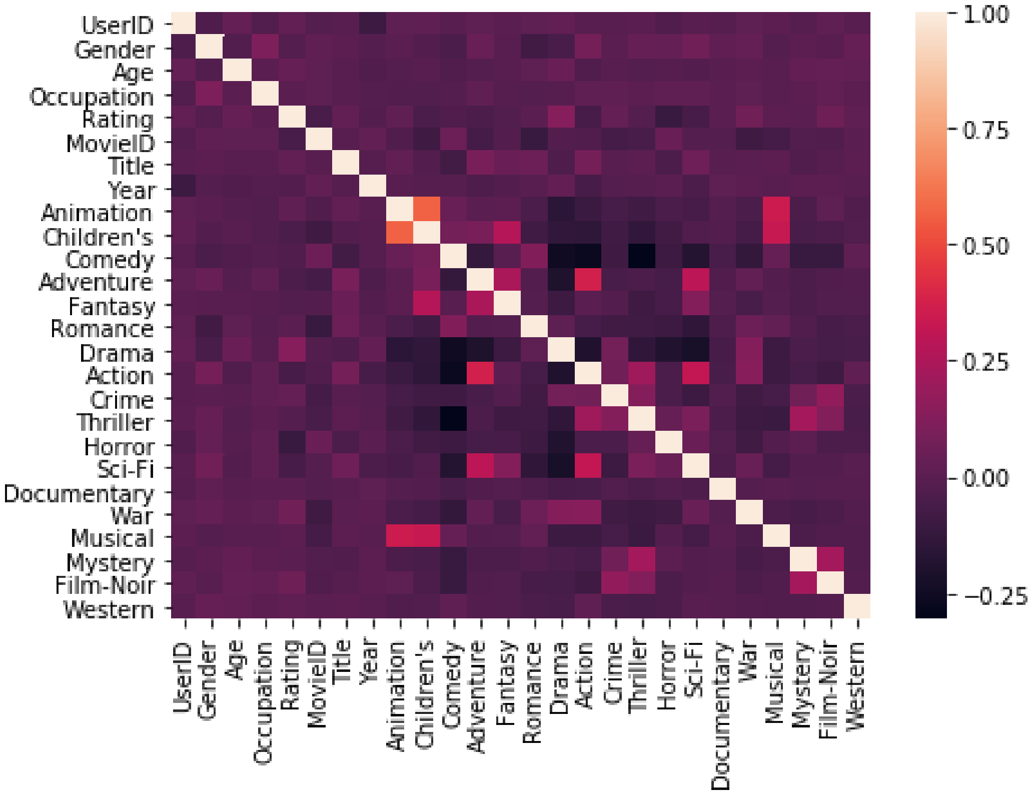
Generally, there is weak correlation between Number of Ratings and Average of Ratings for each movie in all genres with a very small p-value and less than **α** = .01, and small positive correlation between number of ratings and average of ratings with pearson_r 0.36.

After analyze correlation between Number of Ratings and Average of Ratings for 3 populare genres: Comedy, Drama and Action, turns out correlation between these two variables for genre ***Action*** is stronger than the other two genres.

Machine Learning: Overview

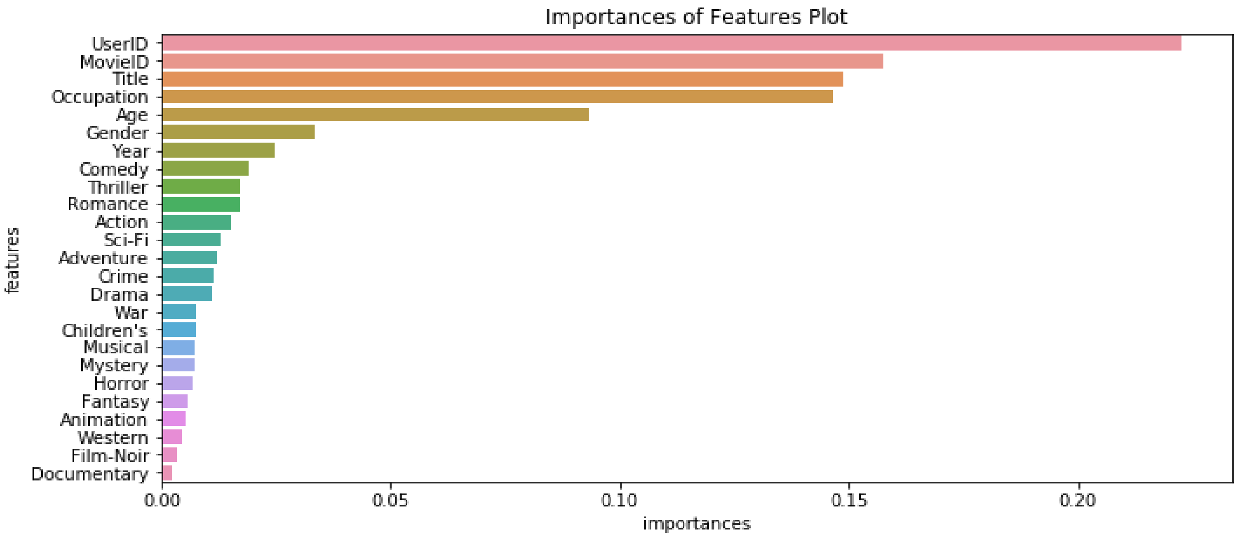
- Random Forest (RF) was used to find out important features and reduce dimensionality
- XGBoost was used for prediction.
- Overall strategy:
    1. Get 1% of data as a sample.
    2. Use Random Forest to find out important components and reduce dimensionality
    3. Use XGBoost on important components of all data.
    4. Split the data into training and test datasets
    5. Train various learning models, with cross-validated hyperparameter tuning
    6. Apply the models to the test dataset to get measures of performance

Correlations between features



Random Forest

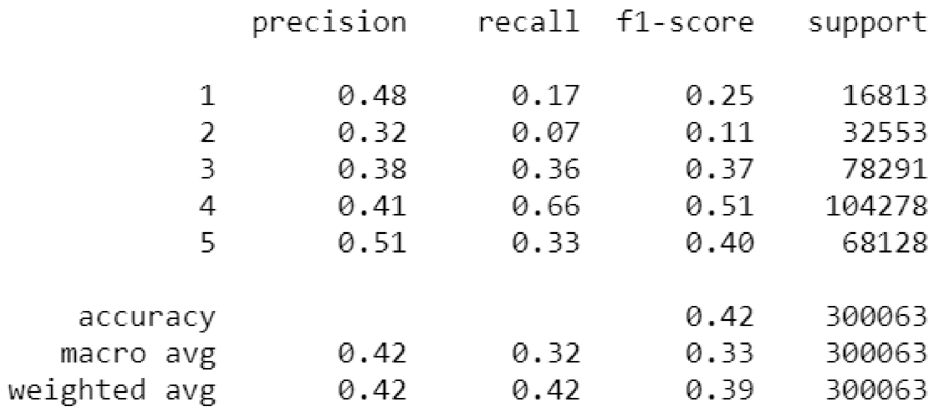
I selected the most informative columns using random forest model.



XGBoost

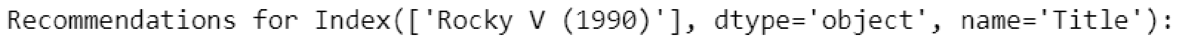
- The best estimator was selected by Randomized Grid Search with Stratified Shuffle Split Cross Validation.
- The best hyperparameters included a tree count of 250, max depth of 10, and allowing trees to consider all features

- **The results of the algorithm**

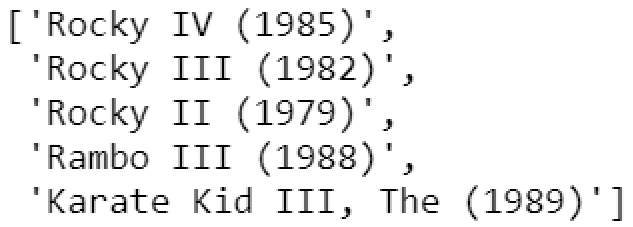


```
              precision    recall  f1-score   support

           1       0.48      0.17      0.25     16813
           2       0.32      0.07      0.11     32553
           3       0.38      0.36      0.37     78291
           4       0.41      0.66      0.51    104278
           5       0.51      0.33      0.40     68128

    accuracy                           0.42    300063
   macro avg       0.42      0.32      0.33    300063
weighted avg       0.42      0.42      0.39    300063
```

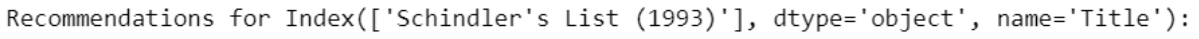
Recommender System



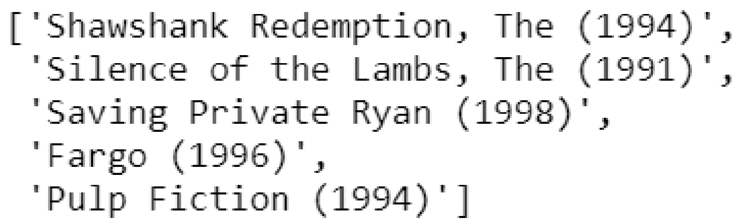
```
Recommendations for Index(['Rocky V (1990)'], dtype='object', name='Title'):
```



```
['Rocky IV (1985)',
 'Rocky III (1982)',
 'Rocky II (1979)',
 'Rambo III (1988)',
 'Karate Kid III, The (1989)']
```



```
Recommendations for Index(['Schindler's List (1993)'], dtype='object', name='Title'):
```



```
['Shawshank Redemption, The (1994)',
 'Silence of the Lambs, The (1991)',
 'Saving Private Ryan (1998)',
 'Fargo (1996)',
 'Pulp Fiction (1994)']
```